AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application.

1-20 (Cancelled)

21. (Currently Amended) A method ~~of pausing processing of instructions,~~ comprising:

determining whether an ~~a first~~ instruction for a first thread is ~~an instruction~~ of a first type ~~at a pipeline stage of a processor~~;

pausing processing of instructions of [[said]] the first thread ~~at said pipeline stage for a period of time if~~ [[said]] upon determining that the ~~first~~ instruction is of [[a]] the first type while processing instructions from a second thread ~~at said pipeline stage~~; and

resuming processing of ~~instructions of said first thread~~ a second microinstruction decoded from the instruction after execution of a first microinstruction decoded from the instruction ~~in response to the determining operation at said pipeline stage~~.

22. (Currently Amended) The method of claim 21 further comprising decoding [[said]] the ~~first~~ instruction into [[a]] the first microinstruction and [[a]] the second microinstruction.

23. (Currently Amended) The method of claim 22 wherein [[said]] the first microinstruction causes [[a.]] a value to be stored in memory for [[said]] the first thread.

24. (Currently Amended) The method of claim 23 further comprising:

processing [[said]] the second microinstruction for execution when [[said]] the value stored in the memory is reset.

25. (Currently Amended) The method of claim 24 wherein [[said]] the value stored in the memory is reset [[if said]] when the first microinstruction is retired.

26. (Currently Amended) A method comprising:

determining~~, at a pipeline stage~~ whether an ~~a first~~ instruction of a first thread is ~~an instruction~~ of a first type;

initiating a counter [[if said]] upon determining that the ~~first~~ instruction is ~~an instruction~~ of the first type, wherein the instruction includes an operand and the initiating includes loading the counter with the operand; and

pausing processing of instructions of [[said]] the first thread ~~at the pipeline stage of a processor~~ until [[said]] the counter reaches [[a]] the ~~predetermined~~ value while processing instructions [[for]] of a second thread at [[said]] the pipeline stage.

27. (Cancelled)

28. (Currently Amended) The method of claim [[27]] 26 further comprising resuming processing instructions of [[said]] the first thread after [[said]] the counter reaches [[said]] the ~~predetermined~~ value.

29. (Currently Amended) An apparatus, comprising:

a decode unit to determine whether ~~a first~~ an instruction of a first thread is ~~an instruction~~ of a first type, said decode unit to pause processing of instructions of said first thread ~~at a pipeline stage of a processor for a period of time if said~~ ~~first~~ upon determining that the instruction is ~~an instruction~~ of the first type by generating a first microinstruction to cause a value to be stored in a memory for the first thread while instructions from a second thread can be processed, said decode unit further to generate a second microinstruction upon which processing is to resume after execution of the first microinstruction ~~cause resumption of processing instructions of said~~

~~first thread in response to the determination at said decode unit~~.

30. (Cancelled)

31. (Cancelled)

32. (Currently Amended) The apparatus of claim [[31]] 29 wherein [[said]] the decode unit is to process [[said]] the second microinstruction ~~if said~~ when the value stored in the memory is reset.

33. (Currently Amended) The apparatus of claim 32 further comprising:

a retire unit coupled to [[said]] the decode unit wherein [[said]] the retire unit is to cause [[said]] the value stored in the memory to be reset ~~if said~~ when the first microinstruction is retired by [[said]] the retire unit.

34. (Currently Amended) An apparatus comprising:

a decode unit to determine whether a ~~first~~ instruction for a first thread is ~~an instruction~~ of a first type;

a counter coupled to [[said]] the decode unit, [[said]] the counter to be loaded with a value of an operand of the instruction ~~initiated~~ if [[said]] the first instruction for [[said]] the first thread is ~~an instruction~~ of [[said]] the first type, [[said]] the decode unit to pause processing instructions of [[said]] the first thread ~~at a pipeline stage of a processor~~ until [[said]] the counter reaches a ~~predetermined~~ the value; and

wherein instructions for a second thread can be processed while instructions of [[said]] the first thread are paused from being processed ~~and wherein said decode unit is to resume processing instructions of said first thread in response to the determination at said decode unit~~.

35. (Cancelled)

36. (Currently Amended) The apparatus of claim [[35]] 34 wherein [[said]] the decode unit ~~can continue~~ is to operate while [[said]] the first thread is paused from being processed.